UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/478,682	01/06/2000	ADAM K. KOLAWA	36463/RRT/P3	1994

23363 7590 04/02/2004

CHRISTIE, PARKER & HALE, LLP
350 WEST COLORADO BOULEVARD
SUITE 500
PASADENA, CA 91105

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2122	13

DATE MAILED: 04/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/478,682	Applicant(s) KOLAWA ET AL.
	Examiner Ted T. Vo	Art Unit 2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the Request Continuation Examination and the amendment filed on 11/26/04. The filing for continued examination under 37 CFR 1.114 is eligible, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114.

Claims 1, 19, 32, and 37 are amended.

Claims 1-37 are pending in the application.

Response to Amendment

2. Applicants' arguments given in their Remarks (pages 9-12) to the amended limitations have been fully considered. The appropriate actions are given below.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-14, 19-27, 32-35, 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cline (US No. 5,313,616).

As per claim 1:

Cline teaching includes 'SBV' and 'DBV' (see column 16, lines 29-34) which scan an input source text to identify procedure calls for generating stubs (see column 16, lines 29-34). The teaching of Cline covers claim limitation:

"***A method for testing a computer program comprising the steps of:***

parsing a source code of the computer program to identify behavior (see Fig. 7) ***of functions in the source code*** (see column 8, lines 54-62, referring to 'SBV' and 'DBV'; that are used to scan the application program. See Fig. 7, a parsing process for identifying behavior, and see column 16, lines 14-34, 'scans the input source text for procedure calls', and 'make a list of all of the calls by location and target');

responsive to the identified behavior of functions, generating stubs to be called by respective functions in the source code for testing the source code (see column 16, lines 35-40);

instrumenting the parsed source code with the generated stubs (see column 16, lines 35-40, and see column 17, lines 56-61);

compiling the instrumented code; testing the compiled code; and reporting test results (see Fig.13, feature 70, see column 17, lines 56-61, see column 19, lines 43-50)";

Cline discloses testing that includes a SBV and a DBV used for parsing an application program to identify the behavior of functions in the application program (See Fig. 7, where Fig. 7 is a known process in compilation that is obtained from parsing a program code to identify the behavior of functions in the program code. The flow graph of basic blocks is typical to the behavior of functions in such program code; where the behavior of a function/procedure/basic block is known as flow graphing of the program code). Responsive to the parsing process in Fig. 7 and scanning for procedure calls in Fig. 14, Cline provides generating the stubs from the input source text (see column 16, lines 29-34). Cline converts the instrumentation with stubs into binary code (see Fig.13, feature 70, see column 17, lines 56-61, see column 19, lines 43-50), for testing the input code.

Cline does not address the source text as source code. However, the difference is the type of inputs implemented in a similar method. The similar method is that, a stub is generated at a file readable/modifiable to human before converting into a form that could not be modifiable. Source code is known as modifiable code and it allows a user to change/correct before compilation. In many aspects, compiled code is in binary form which can't be modifiable. In some aspect, compiled code is in form of assembly language. The code text or object module in the reference that is input to a parser for identifying procedure calls then generating stubs is human readable. This code is also modifiable and compiled once into executable binary form (column 19, lines 43-45). Moreover, Cline suggests a similar method has done at source level (see column 11, lines 44-52). It's obvious that a programmer in both cases (claimed recitation and reference) would conform to a common standard of programming instrumentation; thereby, using the readable and modifiable form (source program) to generate and to instrument the code before being translated into another form that is hard for user's view.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to implement the teaching of Cline for generating stubs at source text in debugging the object code module formed in assembly language into the similar manner of a source program. Doing so would conform to a standard method by using readable and modifiable form to generate and to instrument the code before it becomes another form that is hard from modification.

As per claim 2:

Cline discloses the limitation of claim 2 (See column 18, lines 22-26)

As per claim 3:

Cline discloses the limitation of claim 3; Cline teaches of converting call instructions to call stub locations (column 18, lines 22-26).

As per claim 4:

Cline discloses the limitation of claim 4; Cline teaches generating stubs automatically in using DBV (column 15, lines 42-49).

As per claim 5:

Cline discloses the limitation of claim 5; Cline teaches changing instruction branch (column 18, lines 24-27, lines 42-52).

As per claim 6:

Cline discloses the limitation of claim 6; Cline shows a user-procedure stub in table 6 (column 17).

As per claim 7:

Cline discloses the limitation of claim 6 where GUI is a part of computer system, Cline teaches inherently the claim limitation used in static analysis (see abstract) that require the interaction between user and computer via a GUI.

As per claim 8:

Regarding claim 8, Cline teaches inherently the claim in the teaching of making a list of all of the calls by location and target (see column 16, lines 35-40, or see figure 5).

As per claim 9:

Cline discloses the limitation of claim 19 (referring to procedure, basic blocks; for example see figure 7).

As per claim 10:

Cline discloses the limitation of claim 10 (see Fig. 16b)

As per claim 11:

Cline discloses the limitation of claim 11 (see Fig. 16b)

As per claim 12:

Cline discloses the limitation of claim 12; Cline teaches inherently the claim in maintaining a list of all of the calls by location and target (see column 16, lines 35-40, or see figure 5).

As per claim 13:

Cline discloses the limitation of claim 13; Cline teaches monitoring test coverage of application programs (see column 2, lines 58-60, and figure 12) in using the SBV (see description of SBV, column 9, started from line 43 to column 11 line 62).

As per claim 14:

Cline discloses the limitation of claim 14; Cline discloses the SBV as a tool that has means for testing, monitoring and display (see description of SBV, column 9, started from line 43 to column 11 line 62). SBV has means of providing GUI.

As per claim 19:

Cline teaching includes 'SBV' and 'DBV' (see column 16, lines 29-34) which scan the input source text and identifies procedure calls for generating stubs (see column 16, lines 29-34). The teaching of Cline covers claim limitation:

"***A method for testing a computer program comprising the steps of:***

parsing a source code of the computer program to identify behavior of a plurality of smaller components in the source code (see column 8, lines 54-62, referring to 'SBV' and 'DBV'; that are used to scan the application program. See Fig. 7, a parsing process for identifying behavior, and see column 16, lines 14-34, 'scans the input source text for procedure calls', and 'make a list of all of the calls by location and target');

breaking down the source code into the plurality of smaller components (See Fig. 7, referring to "basic blocks");

based on the identified behavior of plurality of smaller components (See Fig. 9), ***generating stubs to emulate some of the identified behavior of plurality of smaller components individually with the generated stubs emulating some of the identified behavior*** (see column 16, lines 35-40, and see column 17, lines 56-61 and see column 18, lines 21-41, and particularly see Figs. 16a-b, refereeing to the area with "* * *"); ***and***

reporting test results (see column 17, lines 56-61)";

Cline discloses testing that includes a SBV and a DBV used for parsing an application program to identify the behavior of functions in the application program (See Fig. 7, where Fig. 7 is a known process in compilation that is obtained from parsing a program code to identify the behavior of functions in the program code. The flow graph of basic blocks is typical to the behavior of functions in such program code;

where the behavior of a function/procedure/basic block is known as flow graphing of the program code). Responsive to the parsing process in Fig. 7 and scanning for procedure calls in Fig. 14, Cline provides generating the stubs from the input source text (see column 16, lines 29-34). Cline converts the instrumentation with stubs into binary code (see Fig.13, feature 70, see column 17, lines 56-61, see column 19, lines 43-50), for testing the input code.

Cline does not address the source text as source code. However, the difference is the type of inputs implemented in a similar method. The similar method is that, a stub is generated at a file readable/modifiable to human before converting into a form that could not be modifiable. Source code is known as modifiable code and it allows a user to change/correct before compilation. In many aspects, compiled code is in binary form which can't be modifiable. In some aspect, compiled code is in form of assembly language. The code text or object module in the reference that is input to a parser for identifying procedure calls then generating stubs is human readable. This code is also modifiable and compiled once into executable binary form (column 19, lines 43-45). Moreover, Cline suggests a similar method has done at source level (see column 11, lines 44-52). It's obvious that a programmer in both cases (claimed recitation and reference) conform to a common standard of programming instrumentation; thereby, using the readable and modifiable form (source program) to generate and to instrument the code before being translated into another form that is hard for user's view.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to implement the teaching of Cline for generating stubs at source text in debugging the object code module formed in assembly language into the similar manner of a source program. Doing so would conform to a standard method by using readable and modifiable form to generate and to instrument the code before it becomes another form that is hard from modification.

As per claim 20:

Cline discloses the limitation of claim 20 (referring to procedure, basic blocks; for example see figure 7).

As per claim 21:

Cline discloses the limitation of claim 21 (See column 18, lines 22-26)

As per claim 22:

Cline discloses the limitation of claim 22; Cline teaches of converting call instructions to call stub locations (column 18, lines 22-26).

As per claim 23:

Cline discloses the limitation of claim 23; Cline teaches generating stubs automatically in using DBV (column 15, lines 42-49).

As per claim 24:

Cline discloses the limitation of claim 24; Cline teaches changing instruction branch (column 18, lines 24-27, lines 42-52).

As per claim 25:

Cline discloses the limitation of claim 25; Cline shows a user-procedure stub in table 6 (column 17).

As per claim 26:

Cline discloses the limitation of claim 26; Cline teaches monitoring test coverage of application programs (see column 2, lines 58-60, and figure 12) in using the SBV (see description of SBV, column 9, started from line 43 to column 11 line 62).

As per claim 27:

Cline discloses the limitation of claim 27; Cline discloses the SBV as a tool that has means for testing, monitoring and display (see description of SBV, column 9, started from line 43 to column 11 line 62). SBV has means of providing GUI.

As per claim 32:

Claim 32 is a system for testing that recites the claim limitation having claimed functionality corresponding to the method of claim 1. Claim 32 is rejected in the same reason set forth in connecting to the rejection of claim 1.

As per claim 33:

Claim 33 is a system for testing that recites the claim limitation having claimed functionality corresponding to the method of claim 2. Claim 33 is rejected in the same reason set forth in connecting to the rejection of claim 2.

As per claim 34:

Claim 34 is a system for testing that recites the claim limitation having claimed functionality corresponding to the method of claims 8 and claim 9. Claim 34 is rejected in the same reason set forth in connecting to the rejection of claims 8-9.

As per claim 35:

Claim 35 is a system for testing that recites the claim limitation having claimed functionality corresponding to the method of claim 13. Claim 35 is rejected in the same reason set forth in connecting to the rejection of claim 13.

As per claim 37:

Claim 37 is a computer readable medium that recites the claim limitation having claimed functionality corresponding to the method of claim 1. Claim 37 is rejected in the same reason set forth in connecting to the rejection of claim 1.

5. Claims 15, 28, 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cline (US No. 5,313,616) in view of Beizer, "Software Testing Techniques", 1986.

As per claim 15:

Regarding to claim limitation of claim 15, Cline teaches program testing in the manner of the claim 1 as specified above by identifying all call instructions in an application program (see column 16, lines 14-34) and generating stubs and compiling the instrumented application program (see column 16, lines 35-40, column 17, lines 56-61).

Cline does not explicitly mention its program testing comprising steps for defining a specific behavior when a procedure within the source code of the program is called by storing defined information; compiling defined information as a separated object; and linking the compiled object to the code (claimed

limitation: '*defining a specific behavior when a function within the source code is called; storing the defined information; compiling the defined information as a separated object; and linking the compiled object to the code*').

Beizer teaches a basic testing technique in software testing. Beizer teaches that a program behavior is complicated and very hard to understand (section 3.4, page 11). Therefore, it requires building a program model independently (*defined information as a separated object*) from the program to represent the program behavior. It teaches running the built model (*compiling* and *linking*) in order to understand the behavior of the program, and thus it can modify the program (see figure 1-1, page 10). The program model is separated from the program and is run for testing.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to include the teaching of generating program model of Beizer, independently, with the program application testing of Cline. Doing so would conform to the testing standard in which it provides a separated behavior model for easing test works and managements.

As per claim 28:

Claim 28 recites '*The method of claim 19 further comprising the steps of defining a specific behavior when a function within the source code is called; storing the defined information; compiling the defined information as a separated object; and linking the compiled object to the code*". Claim 28, which is further limitation of claims 19, has claimed functionality corresponding to the claimed functionality of claim 15. Claim 28 is rejected in the same reason set forth in connecting to the rejection of claim 15.

As per claim 36:

Claim 36 is a system for testing, which is the further limitation of claims 32, has claimed functionality corresponding to the claimed functionality of claim 15. Claim 36 is rejected in the same reason set forth in connecting to the rejection of claim 15.

6. Claims 16-18, 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cline (US No. 5,313,616).

As per claims 16-18:

Regarding to claim limitation of claims 16: the step of testing comprises white-box testing;

Regarding to claim limitation of claims 17: the step of testing comprises black-box testing;

Regarding to claim limitation of claims 18: the step of testing comprises regression testing:

Cline teaches program testing in the manner of the claim 1 as specified above by identifying all call instructions in an application program (see column 16, lines 14-34) and generating stubs and compiling the instrumented application program (see column 16, lines 35-40, column 17, lines 56-61).

Cline does not explicitly mention its program testing comprising white-box test, black-box test, and regression test.

Official notice is taken that "white-box testing", "black-box testing", and "regression test" are the well-known testing techniques in the testing art. Each kind of these tests is inherent from functional testing or structure testing of the program.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to include each common testing technique at a point's of interest. Doing so would take advantage of all well-known testing techniques that are standardized by the testing art in order to apply each testing technique to a specific testing at points' of interest.

As per claims 29-31:

Claims 29-31, which are further limitation of claims 19, have claimed functionality corresponding to the claimed functionality of claims 16-18 respectively. Claims 29-31 are rejected in the same reason set forth in connecting to the rejection of claims 16-18.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (703) 308-9049. The examiner can normally be

reached on Monday-Friday from 8:00 AM to 5:30 PM ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam, can be reached on (703) 305-4552.

The fax phone numbers:

(703) 872-9306 (for formal communication intended for entry);

(703) 746-5429 (for informal or draft communication, please label "PROPOSED" or "DRAFT").

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

TED T. VO

Patent Examiner
Art Unit: 2122

March 30, 2004